Partial Derivative Computation for Bounded-Impulse Trajectory Models Using Two-Sided Direct Shooting. Part 1: Theory

Donald H. Ellison * and Bruce A. Conway †

University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA

Jacob A. Englander ‡

NASA Goddard Space Flight Center, Greenbelt, MD, 20771, USA

Martin T. Ozimek §

Johns Hopkins University Applied Physics Laboratory, Laurel, MD, 20723, USA

Many optimization methods require accurate partial derivative information in order to ensure efficient, robust, and accurate convergence. This work outlines analytic methods for computing the problem Jacobian for two different bounded-impulse spacecraft trajectory models solved using two-sided shooting. The specific two-body Keplerian propagation method used by both of these models is described. Methods for incorporating realistic operational constraints and hardware models at the preliminary stage of a trajectory design effort are also demonstrated and the analytic methods derived are tested for accuracy using automatic differentiation. A companion paper will solve several relevant problems that show the utility of employing analytic derivatives, i.e. compared to using derivatives found using finite differences.

*Ph. D. Candidate, Department of Aerospace Engineering, 104 South Wright Street, Urbana, IL, Mail Code-236, Student Member AIAA
†Professor Emeritus, Department of Aerospace Engineering, 104 South Wright Street, Urbana, IL, Mail Code-236, Associate Fellow AIAA
‡Aerospace Engineer, Navigation and Mission Design Branch, 8800 Greenbelt Rd, Greenbelt, MD 20771, Member AIAA
§Mission Design Engineer, 11100 Johns Hopkins Road, Laurel, MD, Member AIAA

Nomenclature

$\mathbf{c}$	=	nonlinear program constraint vector
D	=	thruster duty cycle
h_{body}	=	radius of a flyby body
h_{safe}	=	minimum safe flyby altitude
$\mathbf{M}$	=	maneuver transition matrix
m	=	spacecraft mass
N	=	number of phase segments
N_p	=	number of trajectory phases
P	=	power available to the solar electric propulsion unit
$\mathbf{r}$	=	spacecraft position vector w.r.t. central body
r_{flyby}	=	flyby radius (planetary radius + safe flyby altitude)
T	=	thrust
t	=	current epoch
$\mathbf{u}$	=	control vector
$\mathbf{v}$	=	spacecraft velocity vector w.r.t. central body
$\mathbf{x}$	=	nonlinear program decision variable vector
$\mathbf{X}$	=	spacecraft state vector
Δt	=	propagation time of a trajectory segment
Δt_{flight}	=	mission time-of-flight
Δt_p	=	phase time-of-flight
$\boldsymbol{\Theta}$	=	time variable connection matrix
μ	=	standard gravitational parameter
$\boldsymbol{\Xi}$	=	decision variable connection matrix
$\boldsymbol{\Phi}$	=	state transition matrix
$\dot{(\cdot)}$	=	first time derivative
$\ddot{(\cdot)}$	=	second time derivative
$\|\cdot\|$	=	L_2 (Euclidean) norm
$(\cdot)^\dagger$	=	phase match point
$(\cdot)^-$	=	quantity immediately prior to an event
$(\cdot)^+$	=	quantity immediately after an event

$(\cdot)_k$ = impulsive maneuver index

$(\cdot)_{\mathrm{F}}$ = forward propagated half-phase

$(\cdot)_{\mathrm{B}}$ = backward propagated half-phase

$(\cdot)_0$ = phase starting epoch

$(\cdot)_f$ = phase ending epoch

$(\cdot)_{\max}$ = maximum value

$(\cdot)_{\min}$ = minimum value

$(\cdot)_\odot$ = solar quantity measured in the frame of the central body

I. Introduction

The complexity of most spacecraft trajectory design problems necessitates beginning the initial exploration of the design space with lower fidelity models than would be used to create a flight-ready reference solution. Bounded-impulse models are commonly incorporated into preliminary design efforts for both chemical high-thrust [1–7] as well as continuous-thrust propulsion system configurations [8–12]. A variety of optimization techniques have been applied to solve impulsive trajectory optimization problems, however most can be categorized as indirect, direct or hybrid methods. Indirect methods [13–19] use necessary conditions derived using the calculus of variations [20, 21], direct methods [22–27] cast the trajectory optimization problem as a parameter optimization problem and directly extremalize a cost function, and hybrid approaches [28–31] rely on Bellman's Principle of Optimality of a dynamic program. While each method has advantages and disadvantages, direct methods are popular as they are typically quite robust, easily accomadate various problems and constraints, and benefit from the availability of several existing optimization packages such as SNOPT [32], IPOPT [33], and WORHP [34] that may be used to solve the underlying parameter optimization problem, which usually is a nonlinear program.

The necessary conditions for optimality of a nonlinear program [35, 36] require the computation of the Jacobian matrix, that is the matrix of partial derivatives of each of the nonlinear constraints and the objective function with respect to each of the problem decision variables. It is possible to approximate this matrix using the method of finite differences. While implementation of this technique is straighforward, it is computationally expensive and suffers from the competing goals of reducing Taylor series truncation error and floating point roundoff error, which inherently limits the accuracy of this method. Other methods such as automatic differentiation and complex step differentiation offer a means for computing near-machine precision derivatives, however, their implementation can be nontrivial and can also result in a substantial increase in execution time. Lantoine et al. [37] showed that computing state transition matrices (STMs) using multicomplex numbers can result in only marginal computation time increases over calculating the matrices analytically, however this does not preclude having to develop a method for computing the actual constraint partials, towards the calculation of which the STMs play a significant role. When an overloading

technique is integrated into the more general case of the evaluation of the trajectory cost function, it has been shown that derivative computations still rely on additional computational infrastructure such as parallelization of the gradient calculations in order to achieve the necessary runtimes needed to support a preliminary design effort [38].

The increasing sophistication of space missions is driving a similar evolution in preliminary design techniques. Modern hybrid optimal control (HOC) architectures typically require the evaluation of thousands of instances of an NLP cost function and require that the cost function evaluation be accurate, rapid and robust. This is especially true of multi-objective HOC solvers that seek to generate Pareto surfaces [39–41]. The accurate and efficient computation of constraint gradients has been shown to be of critical importance to a preliminary design optimization framework [42, 43]. With this as motivation, the techniques for computing the Jacobian matrix for the multiple gravity assist low-thrust (MGALT) and multiple gravity assist with n deep space maneuvers using shooting (MGAnDSMs) [44] transcriptions will be discussed. A companion paper will solve several relevant problems that show the utility of employing analytic derivatives, i.e. compared to using derivatives computed using finite differences [45].

II. Forward-Backward Shooting Methods

II. A. General Description

The general spacecraft trajectory optimization problem is a multi-phase optimal control problem and a mission trajectory may be organized into N_p phases. The boundaries of a Forward Backward Shooting (FBS) phase are called control points and can be massive bodies (flyby targets) such as planets, their satellites, asteroids, etc., or even a free point in space. The optimizer encodes the mass and the relative velocity vector of the spacecraft with respect to the control points ($\mathbf{v}_\infty$) at either side of the phase as decision variables. These decision parameters are combined with ephemeris data to form the complete spacecraft state at either side of a phase, which is then propagated forwards and backwards from the control points at the phase boundaries. This two-sided shooting, in general results in a discontinuity of the spacecraft's state vector ($\mathbf{X}$) at some location along the phase, which is closed to within some tolerance by the numerical optimizer as shown in Figure 1.

II. B. Constraints and Decision Variables

The hallmark constraint of an FBS phase is the *match point* continuity constraint. In order to enforce continuity of the state vector across the match point, a vector of "defect" constraints is applied, and designated by a dagger (†):

$$\mathbf{c}^\dagger = \mathbf{X}_\mathrm{B}^\dagger - \mathbf{X}_\mathrm{F}^\dagger = \begin{bmatrix} \mathbf{r}_\mathrm{B} - \mathbf{r}_\mathrm{F} \\ \mathbf{v}_\mathrm{B} - \mathbf{v}_\mathrm{F} \\ m_\mathrm{B} - m_\mathrm{F} \end{bmatrix} = \mathbf{0} \tag{1}$$

As previously mentioned, the optimizer is free to select the spacecraft's velocity vector components relative to the two control points defining a particular phase. If the current phase ends with a flyby of a massive body, two nonlinear

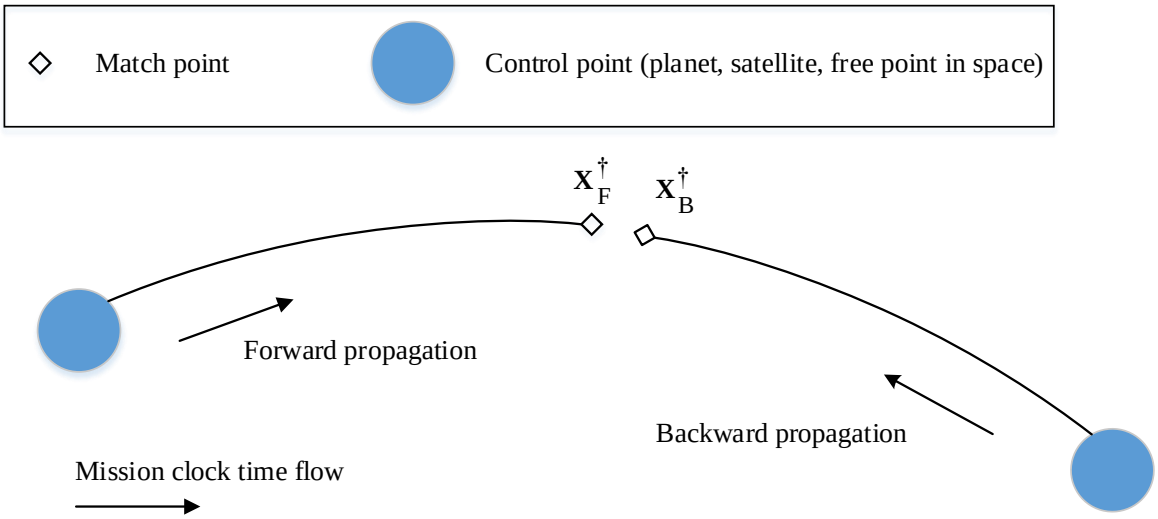

Figure 1. A single FBS phase.

constraints are applied to ensure that the maneuver is feasible, with the flyby itself being modeled as a discontinuity in the spacecraft's velocity vector. The first constraint forces the magnitudes of the incoming and outgoing velocity asymptotes to be equal:

$$c_{v_\infty} = v_\infty^+ - v_\infty^- = 0 \tag{2}$$

The second constraint prevents the altitude of the spacecraft from dropping below a minimum safe altitude h_{safe}, and also ensures that the turn angle of the flyby is physically realizable. We have chosen a safety altitude of 2% of the body's radius h_{body}.

$$\begin{aligned} c_{\text{flyby-altitude}} &= r_{\text{periapse}} - (h_{\text{body}} + h_{\text{safe}}) \geq 0 \\ &= \frac{\mu_{\text{planet}}}{v_\infty^2}\left[\frac{1}{\sin(\delta/2)} - 1\right] - (h_{\text{body}} + h_{\text{safe}}) \geq 0 \end{aligned} \tag{3}$$

where δ is defined in Eq. (4).

$$\delta = \text{acos}\left(\frac{\mathbf{v}_\infty^- \cdot \mathbf{v}_\infty^+}{v_\infty^- \; v_\infty^+}\right) \tag{4}$$

Gradients for Equations (2) and (3) are provided in the Appendix (C).

The last constraint applied to an FBS phase requires that the sum of the individual phase flight times (Δt_p) fall within the upper and lower bounds (Δt_{min} and Δt_{max}) placed on the total mission *time-of-flight* (TOF) Δt_{flight}:

$$c_{\text{TOF}} = \Delta t_{\text{min}} \leq \Delta t_{\text{flight}} = \sum_{i=1}^{N_p} \Delta t_{p_i} \leq \Delta t_{\text{max}} \tag{5}$$

A list of typical decision variables that characterize an FBS mission are described in Table 1.

Table 1. Typical decision variables for an FBS mission.

x_i	Description	Number
t_0	Launch epoch	1
v_∞	Launch impulse magnitude	1
RA	Right ascension of launch asymptote	1
DEC	Declination of launch asymptote	1
Δt_p	Phase time of flight	N_p
m_f	Phase final mass	N_p
$\mathbf{v}_{\infty_0}$	Phase initial excess velocity vector	$3(N_p - 1)$
$\mathbf{v}_{\infty_f}$	Phase final excess velocity vector	$3N_p$ [a]

[a] For a rendezvous this becomes $3(N_p - 1)$

With the concept of an FBS phase established, it is now possible to specialize the concept to two impulsive trajectory models, the multiple gravity assist low-thrust (MGALT) and multiple gravity assist with n deep space maneuvers using shooting (MGAnDSMs) transcriptions.

II. C. Multiple Gravity-Assist Low-Thrust Model

The medium-fidelity continuous-thrust trajectory model used in this work is the multiple gravity assist with low-thrust (MGALT) transcription. This is a simplified model that combines the well-known Sims-Flanagan transcription [8] with a two-body patched-conic flyby model [46]. Each phase of a Sims-Flanagan trajectory is itself a multi-stage optimal control problem, and is discretized into N equal-sized time segments. The continuous-thrust that may be applied over the course of each segment is approximated by a bounded-impulse at the center of each of these segments. Since applied thrust is approximated as a discontinuous $\Delta\mathbf{v}$ vector, it is possible to propagate the spacecraft's position and velocity components using Kepler's equation between applied impulses. Furthermore, due to this discontinuity, the following notation is adopted to distinguish between the spacecraft's velocity just prior to, and immediately following, the applied impulse, i.e.

$$\mathbf{v}_k^+ = \mathbf{v}_k^- + \Delta v_{\max_k} \mathbf{u}_k \tag{6}$$

for forward propagation, and

$$\mathbf{v}_k^- = \mathbf{v}_k^+ - \Delta v_{\max_k} \mathbf{u}_k \tag{7}$$

for backward propagation. Figure 2 depicts a single phase and shows the match point discontinuity. Note that in the backwards propagation half-phase, the $-$ and $+$ superscripts on the pre and post-impulse spacecraft velocities preserve

the manner by which a trajectory would be physically flown and not the way it is numerically propagated in the solver.

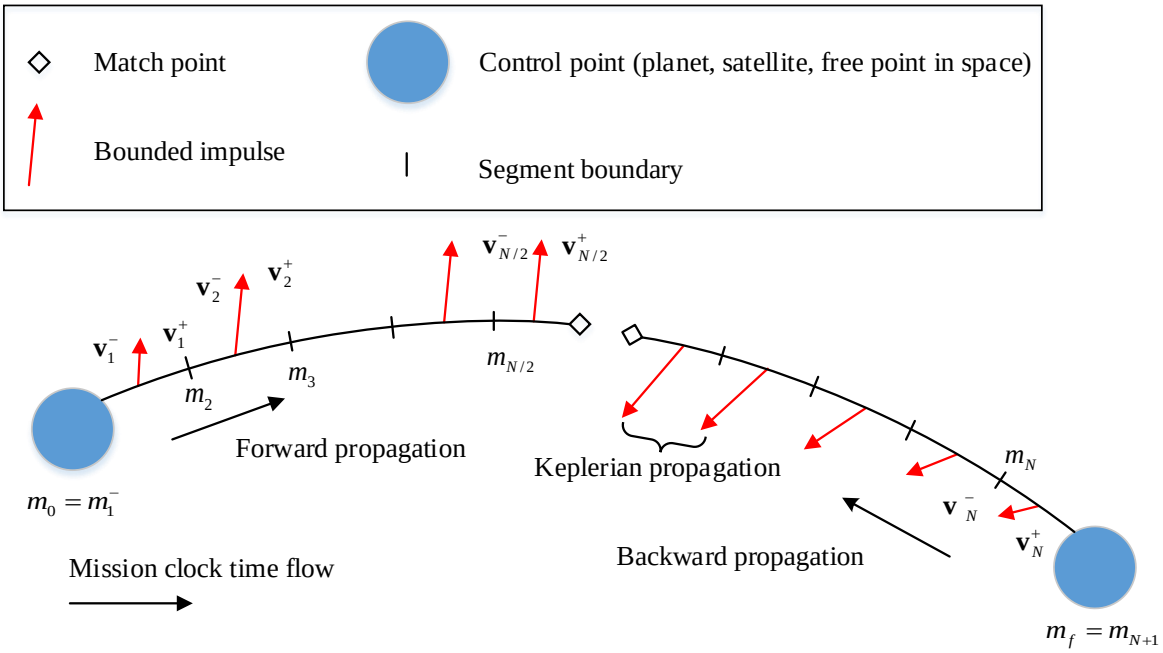

Figure 2. A single MGALT phase. The velocity symbols $\mathbf{v}_k^-$ and $\mathbf{v}_k^+$ are present to indicate the velocity of the spacecraft immediately prior to and immediately after the kth impulsive maneuver.

The 3x1 vector $\mathbf{u}_k$ contains the control parameters associated with the k^{th} maneuver. The scalar quantity $\Delta v_{\max_k}$,

$$\Delta v_{\max_k} = \frac{N_{\text{active}} D\, T_{\max_k} (t_f - t_0)}{m_k N} \tag{8}$$

represents the maximum Δv achievable by the spacecraft by applying the k^{th} maneuver. In Equation (8), N_{active} is the number of active thrusters, D is the thruster duty cycle, $T_{\max_k}$ is the maximum available thrust for the current maneuver, t_0 and t_f are the beginning and ending epochs of the current phase and m_k is the mass of the spacecraft at the center of the segment, just prior to the applied impulse. The spacecraft's mass across the k^{th} bounded impulse is computed using the following equation:

$$m_k = \begin{cases} m_{k-1} - \|\mathbf{u}_{k-1}\|\, \Delta m_{\max_{k-1}} & \text{forward propagation} \\ m_{k+1} + \|\mathbf{u}_k\|\, \Delta m_{\max_k} & \text{backward propagation} \end{cases} \tag{9}$$

where $m_{\max_k} = D\, \Delta t\, \dot{m}_{\max_k}$ and $\Delta t = \frac{(t_f - t_0)}{N}$. Note that the impulsive thrust approximation implies $m_{k-1}^+ = m_k^-$ and for notational convenience, we set $m_k^- = m_k$.

The maximum available thrust $T_{\max}$ and the maximum mass flow rate $\dot{m}_{\max}$ are computed using a propulsion

system power model [45, 47]. These quantities are typically functions of their input power P, which is in turn a function of the spacecraft's distance from the sun $r_{s/\odot}$ for the case of a solar electric power system. In advanced power system models that model hardware degredation, the power available may also be dependent on the time since launch, and will therefore have a direct dependency on the current epoch t i.e.

$$T_{\max_k}(P_k(r_{s/\odot}, t)); \quad \dot{m}_{\max_k}(P_k(t, r_{s/\odot}, t)) \tag{10}$$

II. C. 1. MGALT-Specific Decision Variables and Constraints

An MGALT phase is a specific variety of the general FBS phase. As such, it inherits the constraints and decision variables as described in section II. B. As previously mentioned, the MGALT phase is divided into N equal time segments. The impulse that is applied at the center of each of these steps is comprised of three decision variables that determine the throttle vector $\mathbf{u}_k$, whose norm is bounded by one, i.e.

$$\mathbf{u}_k = \begin{bmatrix} u_{x_k} & u_{y_k} & u_{z_k} \end{bmatrix}^T; \qquad \|\mathbf{u}_k\| \leq 1 \tag{11}$$

represents the "up-to-unit vector control" stage constraint. Additionally, if a variable specific-impulse (VSI) configuration is being approximated, the propulsion unit's specific impulse can be selected for each segment or for the mission as a whole. The decision variables of an MGALT phase (in addition to those in Table 1) are described in Table 2.

Table 2. Additional decision variables for an MGALT phase.

x_i	Description	Number
$\mathbf{u}_k$	control vector	$3N$
I_{sp_k}	specific impulse	N [a]

[a] I_{sp} may also be set for the mission as a whole, in which case only one decision variable is required.

II. D. Multiple Gravity-Assist with n Deep Space Maneuvers Model

The multiple gravity assists (MGA) with n deep space maneuvers per phase, using a shooting technique (DSMs), is an FBS phase that models the trajectory of a spacecraft using a chemical engine [44]. A typical single MGAnDSMs phase is depicted in Figure 3. This phase type allows for the number of mid-course maneuvers to be selected *a priori* or by an outer-loop optimizer. The maneuvers are separated in time by a Δt_k optimization variable that determines their location in the phase. Note that maneuvers at the phase end points are also possible. Alternatively, the phase endpoint may be a gravity-assist body. If fewer than n maneuvers are optimal for the transfer, the formulation is structured such that one or more of the potential maneuvers will have a magnitude of zero.

Unlike MGALT, the applied Δv at each maneuver is encoded directly as a decision variable:

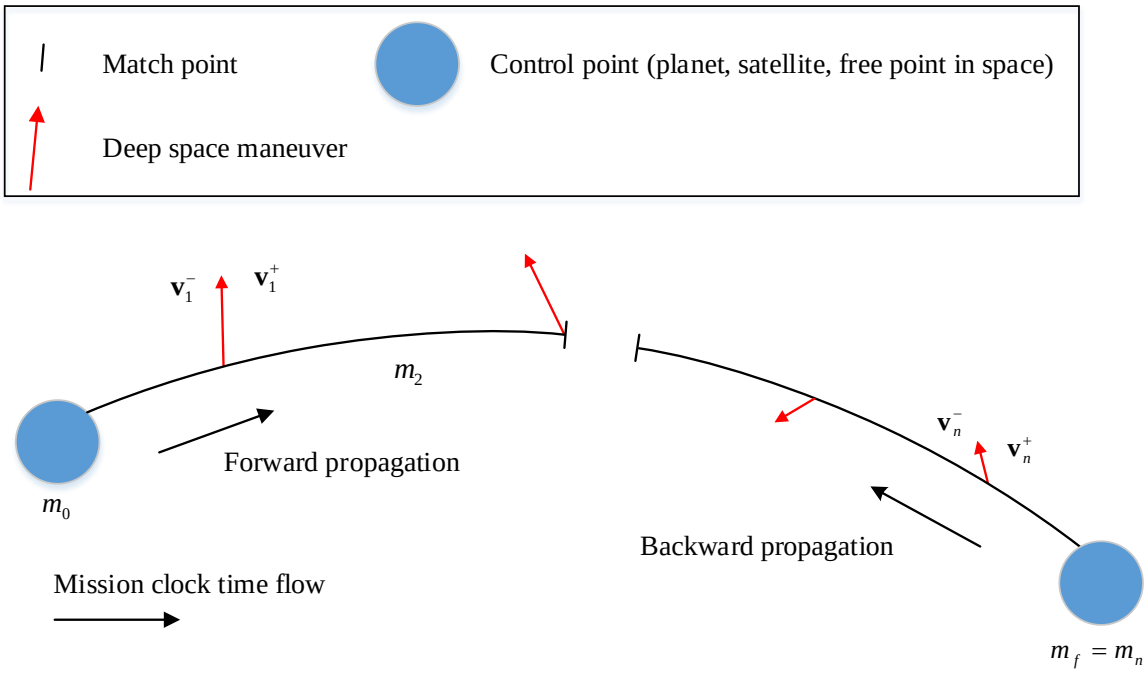

Figure 3. A single MGAnDSMs phase.

$$\mathbf{v}_k^+ = \mathbf{v}_k^- + \Delta\mathbf{v}_k \tag{12}$$

Furthermore, mass is propagated using the Tsiolkovsky rocket equation:

$$m_{k+1} = m_k e^{-\Delta v_k / v_e} \tag{13}$$

where v_e is the rocket's exhaust velocity measured relative to the vehicle.

II. D. 1. MGAnDSMs-Specific Decision Variables and Constraints

As with MGALT, the specialization of an FBS phase to an MGAnDSMs phase introduces several additional optimization parameters and constraints. As summarized in Table 3, for this phase type, the mid-course maneuver vectors (if any), and the relative time variables Δt_k that separate them must be selected by the NLP solver.

Table 3. Additional decision variables for an MGAnDSMs phase.

x_i	Description	Number
Δt_1	time to first maneuver	1
$\Delta t_2, ..., \Delta t_n$	inter-maneuver times	$n-2$
Δt_{n+1}	time from last maneuver to phase end	1
$\Delta \mathbf{v}_1, ..., \Delta \mathbf{v}_n$	DSM vectors	$3n$

The inter-maneuver times Δt_k are constrained such that their sum does not exceed the phase flight time:

$$\sum_{k=1}^{n+1} \Delta t_k - \Delta t_p = 0 \tag{14}$$

In practice, the true decision variable selected by the NLP solver, is $\alpha_k \in [0, 1]$, with $\Delta t_k = \alpha_k \Delta t_p$. Then, the elements in the set $\{\alpha_i\}$ are constrained such that their sum does not exceed one.

III. Match Point Partial Derivatives Computation

III. A. Derivative Propagation

A large majority of the dense entries of the Jacobian matrix for the MGALT and MGAnDSMs transcriptions are comprised of partial derivatives of the phase match points. This is due to the fact that partials of the match point constraint vector $\mathbf{c}^\dagger$ are dense with respect to nearly all entries in the decision vector $\mathbf{x}$. These derivatives are also the most complicated Jacobian entries to calculate. At the most general level, their computation can be summarized by the following expression:

$$\frac{\partial \mathbf{c}^\dagger}{\partial \mathbf{x}} = \frac{\partial \mathbf{X}_\mathrm{B}^\dagger}{\partial \mathbf{x}} - \frac{\mathbf{X}_\mathrm{F}^\dagger}{\partial \mathbf{x}} \tag{15}$$

Calculating the matrix Eq. (15) requires transmitting sensitivity information from various intermediate points along a phase onwards to the match point. State transition matrices (STMs) can be used to map derivatives across Keplerian arcs. The bounded impulse approximation introduces a velocity discontinuity at the location of each maneuver (i.e. $\mathbf{r}(t) \in C^0$ and $\mathbf{v}(t) \in C^{-1}$). For this reason, in addition to the STMs, a method is also required for mapping derivative information across these discontinuities. This mapping can also be expressed as a matrix and is hereafter referred to as the maneuver transition matrix (MTM). The alternating STM/MTM derivative mapping technique is illustrated in Figure 4. It is useful, to associate the maneuver index k with the STMs and MTMs in order to keep track of the large number of calculations required to compute the match point partials.

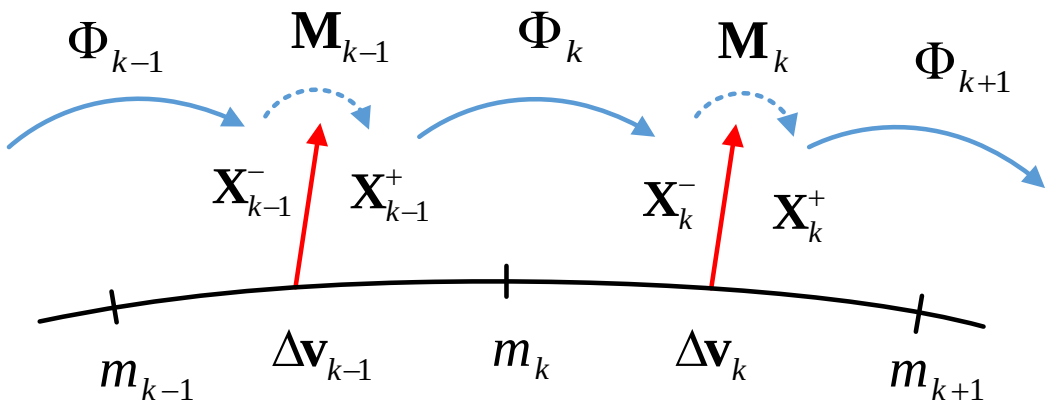

Figure 4. Derivative mapping matrices.

III. A. 1. Two-Body State Transition Matrix

The two-body perturbation state transition matrix contains the first-order sensitivities of the spacecraft's position and velocity vectors at arbitrary time t_k with respect to variations in the position/velocity vector at a previous time t_{k-1}.

$$\mathbf{\Phi}(t_k, t_{k-1}) = \begin{bmatrix} \tilde{\mathbf{R}}(t) & \mathbf{R}(t) \\ \tilde{\mathbf{V}}(t) & \mathbf{V}(t) \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{r}_k}{\partial \mathbf{r}_{k-1}} & \frac{\partial \mathbf{r}_k}{\partial \mathbf{v}_{k-1}} \\ \frac{\partial \mathbf{v}_k}{\partial \mathbf{r}_{k-1}} & \frac{\partial \mathbf{v}_k}{\partial \mathbf{v}_{k-1}} \end{bmatrix} \tag{16}$$

Notational differences abound for the 3x3 quadrants of the perturbation STM [48–53], therefore we included those due to Battin in the Appendix (Equations (97) - (100)).

III. A. 2. Augmented State Transition Matrix

In order to account for first order sensitivities of mass, time-of-flight and specific impulse variables across the k^{th} trajectory segment, the STM is augmented with additional rows and columns:

$$
\begin{aligned}
\mathbf{\Phi}_k &= \frac{\partial \mathbf{X}_k^-}{\partial \mathbf{X}_{k-1}^+} \\
&= \left[\begin{array}{cc|c|cc|c}
\tilde{\mathbf{R}}_k(t) & \mathbf{R}_k(t) & \multirow{2}{*}{$\mathbf{0}_{6\times 1}$} & \frac{\partial \mathbf{r}_k^-}{\partial \Delta t_p} & \multirow{2}{*}{$\mathbf{0}_{6\times 2}$} \\
\tilde{\mathbf{V}}_k(t) & \mathbf{V}_k(t) & & \frac{\partial \mathbf{v}_k^-}{\partial \Delta t_p} & \\
\hline
\multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}_{4\times 6}$}} & \multicolumn{3}{c}{\multirow{2}{*}{$\mathbb{I}_{4\times 4}$}} \\
\multicolumn{2}{c|}{} & \multicolumn{3}{c}{}
\end{array}\right] \\
&= \left[\begin{array}{cc|c|cc|c}
\frac{\partial \mathbf{r}_k^-}{\partial \mathbf{r}_{k-1}^+} & \frac{\partial \mathbf{r}_k^-}{\partial \mathbf{v}_{k-1}^+} & \multirow{2}{*}{$\mathbf{0}_{6\times 1}$} & \frac{\partial \mathbf{r}_k^-}{\partial \Delta t_p} & \multirow{2}{*}{$\mathbf{0}_{6\times 2}$} \\
\frac{\partial \mathbf{v}_k^-}{\partial \mathbf{r}_{k-1}^+} & \frac{\partial \mathbf{v}_k^-}{\partial \mathbf{v}_{k-1}^+} & & \frac{\partial \mathbf{v}_k^-}{\partial \Delta t_p} & \\
\hline
\multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}_{4\times 6}$}} & \multicolumn{3}{c}{\multirow{2}{*}{$\mathbb{I}_{4\times 4}$}} \\
\multicolumn{2}{c|}{} & \multicolumn{3}{c}{}
\end{array}\right]
\end{aligned}
\tag{17}
$$

Proceeding from the left, the first additional row/column in Eq. (17) is for the spacecraft's mass, then current phase time-of-flight, previous phase flight times, and finally I_{sp}. This augmented state transition matrix may be used for both the MGALT and MGAnDSMs transcriptions as both transcriptions use Keplerian two-body propagation methods. For the high-thrust model, the last rows and columns, corresponding to I_{sp} decision variables, are omitted.

III. A. 3. MGALT Maneuver Transition Matrix

Unlike the STM, the MTM is transcription dependent due to the presence of thruster hardware modeling for MGALT and because mass is propagated differently for the two trajectory models. The MGALT MTM is calculated as follows:

$$
\begin{aligned}
\mathbf{M}_k &= \frac{\partial \mathbf{X}_k^+}{\partial \mathbf{X}_k^-} \\
&= \left[\begin{array}{cc|c|cc|c}
\frac{\partial \mathbf{r}_k^+}{\partial \mathbf{r}_k^-} & \frac{\partial \mathbf{r}_k^+}{\partial \mathbf{v}_k^-} & \frac{\partial \mathbf{r}_k^+}{\partial m_k} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\
\frac{\partial \mathbf{v}_k^+}{\partial \mathbf{r}_k^-} & \frac{\partial \mathbf{v}_k^+}{\partial \mathbf{v}_k^-} & \frac{\partial \mathbf{v}_k^+}{\partial m_k} & \mathbf{M}_{k_{24}} & \mathbf{0}_{3\times1} & \mathbf{M}_{k_{26}} \\
\hline
\frac{\partial m_{k+1}}{\partial \mathbf{r}_k^-} & \mathbf{0}_{1\times3} & 1 & \mathbf{M}_{k_{34}} & 0 & \mathbf{M}_{k_{36}} \\
\hline
\multicolumn{3}{c|}{\mathbf{0}_{3\times7}} & \multicolumn{3}{c}{\mathbb{I}_{3\times3}}
\end{array}\right] \\
&= \left[\begin{array}{cc|c|cc|c}
\mathbb{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\
\frac{\partial \mathbf{v}_k^+}{\partial \mathbf{r}_k^-} & \mathbb{I}_{3\times3} & \frac{\partial \mathbf{v}_k^+}{\partial m_k} & \mathbf{M}_{k_{24}} & \mathbf{0}_{3\times1} & \mathbf{M}_{k_{26}} \\
\hline
\frac{\partial m_{k+1}}{\partial \mathbf{r}_k^-} & \mathbf{0}_{1\times3} & 1 & \mathbf{M}_{k_{34}} & 0 & \mathbf{M}_{k_{36}} \\
\hline
\multicolumn{3}{c|}{\mathbf{0}_{3\times7}} & \multicolumn{3}{c}{\mathbb{I}_{3\times3}}
\end{array}\right]
\end{aligned}
\tag{18}
$$

For forward propagated half-phases:

$$
\frac{\partial \mathbf{v}_k^+}{\partial \mathbf{r}_k^-} = \frac{\partial \Delta v_{\max_k}}{\partial \mathbf{r}_k^-} = \mathbf{u}_k \frac{D\Delta t}{m_k} \cdot \frac{\partial T_{\max_k}}{\partial P_k} \cdot \frac{\partial P_k}{\partial r_{s/\odot_k}} \cdot \frac{\partial r_{s/\odot_k}}{\partial \mathbf{r}_k} \tag{19}
$$

Note here that the last term in Eq. (19) implies that these calculations are valid for any central body, i.e.,

$$
\mathbf{r}_{s/\odot} = \mathbf{r} - \mathbf{r}_{\odot} \tag{20}
$$

and

$$
\frac{\partial r_{s/\odot_k}}{\partial \mathbf{r}_k} = \left[\frac{\partial r_{s/\odot}}{\partial x_k} \;\; \frac{\partial r_{s/\odot}}{\partial y_k} \;\; \frac{\partial r_{s/\odot}}{\partial z_k}\right] \tag{21}
$$

This sub-matrix is similarly calculated for backwards propagated half-phases, with the addition of an extra term that accounts for the fact that in Eq. (9), m_k has a direct dependence on $\dot{m}_{max_k}$ (and also noting that $\mathbf{r}_k^+ = \mathbf{r}_k^- = \mathbf{r}_k$):

$$
\begin{aligned}
\frac{\partial \mathbf{v}_k^-}{\partial \mathbf{r}_k^+} = \mathbf{u}_k \frac{D\Delta t}{m_k} \cdot \frac{\partial P_k}{\partial r_{s/\odot_k}} \cdot \frac{\partial r_{s/\odot_k}}{\partial \mathbf{r}_k} \cdot \Bigg[& \frac{\partial T_{\max_k}}{\partial P_k} \\
& - \|\mathbf{u}_k\| \frac{D\Delta t T_{\max_k}}{m_k} \cdot \frac{\partial \dot{m}_k}{\partial P_k} \Bigg]
\end{aligned}
\tag{22}
$$

From Eq. (6) and (7), it follows that:

$$\frac{\partial \mathbf{v}_k^+}{\partial \mathbf{m}_k} = \frac{\partial \Delta v_{\max_k}}{\partial \mathbf{m}_k} = -\mathbf{u}_k \frac{D\Delta t T_{\max_k}}{m_k^2} = -\mathbf{u}_k \frac{\Delta v_{\max_k}}{m_k} \tag{23}$$

$$\frac{\partial \mathbf{v}_k^-}{\partial \mathbf{m}_{k+1}} = \frac{\partial \Delta v_{\max_k}}{\partial \mathbf{m}_k} = \mathbf{u}_k \frac{D\Delta t T_{\max_k}}{m_k^2} = \mathbf{u}_k \frac{\Delta v_{\max_k}}{m_k} \tag{24}$$

The sensitivity of the post/pre-burn mass to changes in the spacecraft's current position is calculated as follows:

$$\frac{\partial m_{k+1}}{\partial \mathbf{r}_k^-} = -\|\mathbf{u}_k\| D\Delta t \, \frac{\partial \dot{m}_k}{\partial P_k} \cdot \frac{\partial P_k}{\partial r_{s/\odot_k}} \cdot \frac{\partial r_{s/\odot_k}}{\partial \mathbf{r}_k} \tag{25}$$

$$\frac{\partial m_k}{\partial \mathbf{r}_k^+} = \|\mathbf{u}_k\| D\Delta t \, \frac{\partial \dot{m}_k}{\partial P_k} \cdot \frac{\partial P_k}{\partial r_{s/\odot_k}} \cdot \frac{\partial r_{s/\odot_k}}{\partial \mathbf{r}_k} \tag{26}$$

The submatrices $\mathbf{M}_{k_{24}}$ and $\mathbf{M}_{k_{34}}$ will be discussed in section III. D and $\mathbf{M}_{k_{26}}$ and $\mathbf{M}_{k_{36}}$ in section III. C.

III. A. 4. MGAnDSMs Maneuver Transition Matrix

The MTM changes slightly for the high-thrust case as previously mentioned since mass is propagated using the rocket equation, and because chemical motors are not typically capable of varying their specific impulse. The MTM becomes:

$$\mathbf{M}_k = \frac{\partial \mathbf{X}_k^+}{\partial \mathbf{X}_k^-} = \left[\begin{array}{ccc|c} \mathbb{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \\ \mathbf{0}_{3\times3} & \mathbb{I}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{7\times2} \\ \hline \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{M}_{k_{33}} & \\ \hline & \mathbf{0}_{7\times2} & & \mathbb{I}_{2\times2} \end{array}\right] \tag{27}$$

where,

$$\mathbf{M}_{k_{33}} = \frac{\partial m_{k+1}}{\partial m_k} = e^{-\Delta v_k / v_e}. \tag{28}$$

III. A. 5. STM-MTM Chain

Once $\mathbf{\Phi}_k$ and $\mathbf{M}_k$ have been calculated for each segment, propagation of derivative information from any point along the trajectory onward to the match point is achieved through sequential multiplication of these matrices:

$$\frac{\partial \mathbf{X}_F^\dagger}{\partial \mathbf{X}_k^+} = \mathbf{\Phi}_{N/2+1} \mathbf{M}_{N/2} \mathbf{\Phi}_{N/2} \cdot \ldots \cdot \mathbf{M}_{k+1} \mathbf{\Phi}_{k+1} \tag{29}$$

The only remaining step towards obtaining the actual derivatives of interest in Eq. (15) is to compute the derivative

$\frac{\partial \mathbf{X}_k^+}{\partial x_i}$, that is the sensitivity of the spacecraft's state immediately following the k^{th} impulse to changes in the i^{th} element of the decision vector. One or more of these derivatives may be contained in the derivative connection matrix $\mathbf{\Xi}_k$. Details on the calculation of $\mathbf{\Xi}_k$ vary depending on the particular p_i being considered and are provided in subsequent sections for several decision vector entries. After computing the derivative connection matrix, the match point Jacobian entries may be calculated as follows:

$$\mathbf{\Phi}_{N/2+1}\mathbf{M}_{N/2}\mathbf{\Phi}_{N/2}\cdot\ldots\cdot\mathbf{M}_{k+1}\mathbf{\Phi}_{k+1}\mathbf{\Xi}_k \tag{30}$$

III. B. Partials With Respect to Segment Control Variables

For both bounded impulse models, the majority of the problem decision variables are those defining the magnitude and the direction of thrust applied over the course of each segment. The MTM submatrices corresponding to these variables are computed differently for the two models.

III. B. 1. MGALT

For the low-thrust case, the actual decision variable is the "up-to-unit" throttle vector that scales the maximum allowed Δv for the segment. For forward propagated half-phases, from Eq. (6):

$$\frac{\partial \mathbf{v}_k^+}{\partial \mathbf{u}_k} = \Delta v_{\max_k}\mathbb{I}_{3\times 3} \tag{31}$$

and from Eq. (9):

$$\frac{\partial m_{k+1}}{\partial \mathbf{u}_k} = -\frac{\mathbf{u}_k^T}{\|\mathbf{u}_k\|} D\,\Delta t\,\dot{m}_{\max_k} \tag{32}$$

therefore,

$$\mathbf{\Xi}_k = \frac{\partial \mathbf{X}_k^+}{\partial \mathbf{u}_k} = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \hline \frac{\partial \mathbf{v}_k^+}{\partial \mathbf{u}_k} \\ \hline \frac{\partial m_{k+1}}{\partial \mathbf{u}_k} \end{bmatrix} \tag{33}$$

and hence,

$$\frac{\partial \mathbf{X}_F^\dagger}{\partial \mathbf{u}_k} = \begin{cases} \mathbf{\Phi}_{k+1}\mathbf{\Xi}_k & k = N/2 \\ \prod_{i=1}^{N/2-k}\left(\mathbf{\Phi}_{N/2+2-i}\mathbf{M}_{N/2+1-i}\right)\mathbf{\Phi}_{k+1}\mathbf{\Xi}_k & 1 \le k < N/2 \end{cases} \tag{34}$$

If we now consider derivatives of the match point constraint with respect to control parameters in the backwards propagated half-phase, the method changes slightly and we are now interested in calculating $\frac{\partial \mathbf{X}_k^-}{\partial \mathbf{u}_k}$:

$$\frac{\partial \mathbf{v}_k^-}{\partial \mathbf{u}_k} = \frac{D\Delta t\, \dot{m}_{\max_k}}{\|\mathbf{u}_k\|} \Delta v_{\max_k} \mathbf{u}_k \mathbf{u}_k^T - \Delta v_{\max_k} \mathbb{I}_{3\times 3} \tag{35}$$

and from Eq. (9):

$$\frac{\partial m_{k+1}}{\partial \mathbf{u}_k} = \frac{\mathbf{u}_k^T}{\|\mathbf{u}_k\|} D\ \Delta t\ \dot{m}_{\max_k} \tag{36}$$

therefore,

$$\mathbf{\Xi}_k = \frac{\partial \mathbf{X}_k^-}{\partial \mathbf{u}_k} = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \hline \frac{\partial \mathbf{v}_k^-}{\partial \mathbf{u}_k} \\ \hline \frac{\partial m_{k+1}}{\partial \mathbf{u}_k} \end{bmatrix} \tag{37}$$

and hence,

$$\frac{\partial \mathbf{X}_B^\dagger}{\partial \mathbf{u}_k} = \begin{cases} \mathbf{\Phi}_{k+1} \mathbf{\Xi}_k & k = N/2+1 \\ \prod_{i=k}^{N} \left(\mathbf{\Phi}_i \mathbf{M}_{i-1} \right) \mathbf{\Phi}_{k+1} \mathbf{\Xi}_k & N/2+1 < k \leq N \end{cases} \tag{38}$$

III. B. 2. MGAnDSMs

Computing match point control sensitivities for MGAnDSMs is done differently from the procedure for the MGALT case in that the applied Δv at each mid-course maneuver is directly encoded as a decision variable (in lieu of the MGALT throttle parameter) and mass is propagated with the rocket equation. Therefore, for forward propagation:

$$\frac{\partial m_{k+1}}{\partial \Delta \mathbf{v}_k} = -\frac{m_k}{v_e} \frac{\Delta \mathbf{v}_k^T}{\Delta v_k} e^{-\Delta v_k / v_e} \tag{39}$$

therefore,

$$\mathbf{\Xi}_k = \frac{\partial \mathbf{X}_k^+}{\partial \Delta \mathbf{v}_k} = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \hline \mathbb{I}_{3\times 3} \\ \hline \frac{\partial m_{k+1}}{\partial \mathbf{u}_k} \end{bmatrix} \tag{40}$$

The calculations are similar for backwards propagation.

The match point derivative calculations for phase final mass and the components of $\mathbf{v}_\infty$ are not discussed in this paper for the sake of brevity, but their computation is analogous to the control sensitivity calculations for both MGALT and MGAnDSMs.

III. C. Partials with Respect to Segment Specific Impulse Variables

Match point sensitivities with respect to I_{sp} (for MGALT) are considered for thruster models operating under two paradigms. The first has the optimizer select the engine's I_{sp} for the entire duration of the mission, which could include multiple phases. The second allows the optimizer to select the I_{sp} for each trajectory segment, modeling a VSI-capable thruster configuration. For both of these cases, it is assumed that the engine's thrust is determined from the following relationship:

$$T_{\max_k} = \frac{2\eta P_k}{I_{\text{sp}} g_0} \tag{41}$$

and the mass flow rate is then computed with:

$$\dot{m}_{\max_k} = \frac{T_{\max_k}}{I_{\text{sp}} g_0} \tag{42}$$

For the first case, the partial derivative computation is accounted for in each MTM by the sub-matrices $\mathbf{M}_{26}$ and $\mathbf{M}_{36}$ from Eq. (18):

$$\mathbf{M}_{26} = \pm \mathbf{u}_k \frac{D\Delta t}{m_k} \left(\frac{\partial T_{\max_k}}{\partial I_{\text{sp}}} - \frac{T_{\max_k}}{m_k} \frac{\partial m_k}{\partial I_{\text{sp}}} \right) \tag{43}$$

where

$$\mathbf{M}_{36} = \frac{\partial m_k}{\partial I_{\text{sp}}} = \mp \|\mathbf{u}_k\| D\Delta t \frac{\partial \dot{m}_{\max_k}}{\partial I_{\text{sp}}} \tag{44}$$

For a VSI configuration where the optimizer is free to select the I_{sp} for each phase segment, the match point derivatives are computed in a similar fashion to the control derivatives. It is worth noting that regarding actual thruster hardware, I_{sp} is not a directly selectable quantity. Both specific impulse and thrust are varied by discretely altering the input voltage and the mass flow rate to the thruster.The STM-MTM chains used in Eq. (34) and (38) are used to compute the VSI derivatives, with only the Ξ_k vector changing for the case of segment I_{sp}:

$$\mathbf{\Xi}_k = \frac{\partial \mathbf{X}_k^+}{\partial I_{\text{sp}_k}} = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \hline \frac{\partial \mathbf{v}_k^+}{\partial \mathbf{u}_k} \\ \hline \frac{\partial m_{k+1}}{\partial \mathbf{u}_k} \end{bmatrix} \tag{45}$$

III. D. Partials with Respect to the Current Phase Flight Time

III. D. 1. MGALT

The current phase flight time variable enters into the match point derivative computations via the STM in addition to the MTM. The STM contains explicit time dependencies of the pre-impulse position and velocity because the

Kepler propagation length of each segment is computed from the current phase flight time. The method for computing these derivatives is due to Pitkin [54] and is also used by Lantoine and Russell for computing state sensitivities with respect to time-of-flight variables across one Keplerian arc [55]. The plus and minus signs correspond to forward and backward propagated half-phases respectively. Expressions for $\dot{F}$, $\dot{G}$, $\ddot{F}$ and $\ddot{G}$ are provided in the Appendix (A).

$$\frac{\partial \mathbf{r}_k^-}{\partial \Delta t_p} = \frac{\partial \mathbf{r}_k^-}{\partial \Delta t} \cdot \frac{\partial \Delta t}{\partial \Delta t_p} = \pm \left[\dot{F}\mathbf{r}_{k-1}^+ + \dot{G}\mathbf{v}_{k-1}^+ \right] \frac{\partial \Delta t}{\partial \Delta t_p} \tag{46}$$

$$\frac{\partial \mathbf{v}_k^-}{\partial \Delta t_p} = \frac{\partial \mathbf{v}_k^-}{\partial \Delta t} \cdot \frac{\partial \Delta t}{\partial \Delta t_p} = \pm \left[\ddot{F}\mathbf{r}_{k-1}^+ + \ddot{G}\mathbf{v}_{k-1}^+ \right] \frac{\partial \Delta t}{\partial \Delta t_p} \tag{47}$$

The derivative of segment propagation time with respect to Δt_p is calculated differently for the half-segments than it is for full segments. In the MGALT model, half-segments occur at the phase left and right phase boundaries, and on either side of the match point:

$$\frac{\partial \Delta t}{\partial \Delta t_p} = \begin{cases} \frac{1}{N} & \text{for full segments} \\ \frac{1}{2N} & \text{for half-segments} \end{cases} \tag{48}$$

The MTM also contains entries that facilitate the computation of partial derivatives with respect to Δt_p. Specifically, changes in $\Delta v_{\max_k}$ and $\Delta m_{\max_k}$ due to variations in Δt_p are encoded in each MTM. The submatrices $\mathbf{M}_{k_{24}}$ and $\mathbf{M}_{k_{34}}$ are comprised of Δt_p derivative directions of P_k, Δt and t_k that do not involve the position of the spacecraft at the point of the applied maneuver $\mathbf{r}_k^-$. To see what is meant by this for P_k in particular, it is helpful to look at its partial derivative with respect to Δt_p:

$$\frac{\partial P_k}{\partial \Delta t_p} = \frac{\partial P_k}{\partial r_{s/\odot_k}} \cdot \frac{\partial r_{s/\odot_k}}{\partial \mathbf{r}_k^-} \cdot \frac{\partial \mathbf{r}_k^-}{\partial \Delta t_p} + \frac{\partial P_k}{\partial t_k} \cdot \frac{\partial t_k}{\partial \Delta t_p} \tag{49}$$

The first term in Eq. (49) is accounted for by Equations (19), (22) and (46). The second term must be accounted for by $\mathbf{M}_{k_{24}}$ and $\mathbf{M}_{k_{34}}$.

The sub-matrix $\mathbf{M}_{k_{24}}$ is calculated as follows for forward propagation, where t is the current mission epoch (measured from launch):

$$\mathbf{M}_{k_{24}} = \mathbf{u}_k \frac{D}{m_k} \left[\Delta t \frac{\partial T_{\max_k}}{\partial P_k} \cdot \frac{\partial P_k}{\partial t_k} \cdot \frac{\partial t_k}{\partial \Delta t_p} + \frac{\partial \Delta t}{\partial \Delta t_p} T_{\max_k} \right] \tag{50}$$

and with the added terms accounting for the dependence on $\dot{m}_{\max_k}$ by $\Delta v_{\max_k}$ for backwards propagation.

$$\mathbf{M}_{k_{24}} = -\mathbf{u}_k \frac{D}{m_k}\left[\Delta t \frac{\partial T_{\max_k}}{\partial P_k}\cdot\frac{\partial P_k}{\partial t_k}\cdot\frac{\partial t_k}{\partial \Delta t_p} + \frac{\partial \Delta t}{\partial \Delta t_p} T_{\max_k}\right.$$

$$\left. - \|\mathbf{u}_k\| \frac{D\Delta t\, T_{\max_k}}{m_k}\left(\frac{\dot{m}_k}{N} + \Delta t\, \frac{\partial \dot{m}_{\max_k}}{\partial P_k}\cdot\frac{\partial P_k}{\partial t_k}\cdot\frac{\partial t_k}{\partial \Delta t_p}\right)\right] \tag{51}$$

The derivative of the current epoch t_k with respect to Δt_p is readily calculated once one considers how the current epoch is computed for a forward propagated half-phase (where t_0 is the epoch at the start of the phase):

$$\begin{aligned} t_k &= t_0 + \Delta t_1 + \Delta t_2 + \Delta t_3 + \ldots + \Delta t_k \\ &= t_0 + \frac{\Delta t_p}{2N} + \frac{\Delta t_p}{N} + \frac{\Delta t_p}{N} + \ldots + \frac{\Delta t_p}{N} \\ &= t_0 + \frac{(k-0.5)}{N}\Delta t_p \end{aligned} \tag{52}$$

thus,

$$\frac{\partial t_k}{\partial \Delta t_p} = \left(\frac{k-0.5}{N}\right) \tag{53}$$

For a backwards propagated half-phase, the calculation is similar and the derivative is:

$$\frac{\partial t_k}{\partial \Delta t_p} = \left(\frac{0.5-k}{N}\right) \tag{54}$$

The final sub-matrix in the augmented MTM is computed as follows, again with the negative sign corresponding with forward propagation, the positive sign with backward:

$$\mathbf{M}_{k_{34}} = \mp\|\mathbf{u}_k\| D\left(\frac{\dot{m}_{\max_k}}{N} + \Delta t\, \frac{\partial \dot{m}_{\max_k}}{\partial P_k}\cdot\frac{\partial P_k}{\partial t_k}\cdot\frac{\partial t_k}{\partial \Delta t_p}\right) \tag{55}$$

Then, to actually compute the match point Δt_p gradients, the following matrix multiplications must be carried out with the STM and MTM augmented with Δt_p derivative information:

$$\frac{\partial \mathbf{X}_F^\dagger}{\partial \Delta t_p} = \mathbf{\Phi}_{N/2+1}\mathbf{M}_{N/2}\mathbf{\Phi}_{N/2}\cdot\ldots\cdot\mathbf{\Phi}_2\mathbf{M}_1\mathbf{\Theta}_1 \tag{56}$$

where

$$\boldsymbol{\Theta}_1 = \begin{bmatrix} \frac{\partial \mathbf{r}_1^-}{\partial \Delta t_p} \\ \frac{\partial \mathbf{v}_1^-}{\partial \Delta t_p} \\ 0 \\ 1 \end{bmatrix} \tag{57}$$

and

$$\frac{\partial \mathbf{X}_B^\dagger}{\partial \Delta t_p} = \boldsymbol{\Phi}_{N/2+2}\mathbf{M}_{N/2+1}\boldsymbol{\Phi}_{N/2+3} \cdot \ldots \cdot \boldsymbol{\Phi}_{N+1}\mathbf{M}_N\boldsymbol{\Theta}_N \tag{58}$$

where

$$\boldsymbol{\Theta}_N = \begin{bmatrix} \frac{\phi_B}{0} \\ 1 \end{bmatrix} \tag{59}$$

In Eq. (59), ϕ accounts for the fact that the state at the right hand boundary of a phase has non-zero Δt_p gradients:

$$\phi_B = \begin{bmatrix} \tilde{\mathbf{R}}_{N+2} & \mathbf{R}_{N+2} \\ \tilde{\mathbf{V}}_{N+2} & \mathbf{V}_{N+2} \end{bmatrix} \frac{\partial \mathbf{X}_f}{\partial \Delta t_p} + \begin{bmatrix} \frac{\partial \mathbf{r}_N^+}{\partial \Delta t_p} \\ \frac{\partial \mathbf{v}_N^+}{\partial \Delta t_p} \end{bmatrix} \tag{60}$$

III. D. 2. MGAnDSMs

Computing current phase flight time derivatives for the high thrust transcription requires that Eq. (48) be replaced by:

$$\frac{\partial \Delta t_k}{\partial \Delta t_p} = \alpha_k \tag{61}$$

This is because, for MGAnDSMs, the inter-maneuver Keplerian propagation times Δt_k are not uniform as they are for MGALT, rather each is constructed out of two decision variables the α_k and Δt_p:

III. E. Partials with Respect to Previous Phase Flight Times and Launch Epoch

Flight time variables from previous phases as well as the launch epoch variable affect the state at the left and right boundaries of the current phase. This means that Eq. (57) must be modified:

$$\boldsymbol{\Theta}_1 = \begin{bmatrix} \frac{\phi_F}{0} \\ 1 \end{bmatrix} \tag{62}$$

$$\phi_F = \begin{bmatrix} \tilde{\mathbf{R}}_1 & \mathbf{R}_1 \\ \tilde{\mathbf{V}}_1 & \mathbf{V}_1 \end{bmatrix} \frac{\partial \mathbf{X}_0}{\partial \Delta t_p} + \begin{bmatrix} \frac{\partial \mathbf{r}_1^-}{\partial \Delta t_p} \\ \frac{\partial \mathbf{v}_1^-}{\partial \Delta t_p} \end{bmatrix} \tag{63}$$

Previous phase flight times do not impact the Keplerian propagation time between maneuvers in the current phase. For this reason, the STM does not contain explicit partial derivatives of the state with respect to previous times-of-flight variables.

III. F. Partials with Respect to the MGAnDSMs Inter-Maneuver Times

The match point derivatives with respect to the MGAnDSMs inter-maneuver time variables are computed similarly to the phase flight time variables, except that each one influences a different amount of the half-phase.

$$\frac{\partial \mathbf{X}_F^\dagger}{\partial \Delta t_k} = \mathbf{M}_{N/2} \mathbf{\Phi}_{N/2} \cdot \ldots \cdot \mathbf{\Phi}_{k+1} \mathbf{M}_k \mathbf{\Theta}_k \tag{64}$$

where

$$\mathbf{\Theta}_k = \begin{bmatrix} \frac{\partial \mathbf{r}_k^-}{\partial \Delta t_k} \\ \frac{\partial \mathbf{v}_k^-}{\partial \Delta t_k} \\ 0 \\ 1 \end{bmatrix} \tag{65}$$

$$\frac{\partial \mathbf{X}_B^\dagger}{\partial \Delta t_k} = \mathbf{\Phi}_{N/2+1} \mathbf{M}_{N/2+1} \mathbf{\Phi}_{N/2+2} \cdot \ldots \cdot \mathbf{\Phi}_k \mathbf{M}_k \mathbf{\Theta}_k \tag{66}$$

where

$$\mathbf{\Theta}_k = \begin{bmatrix} \frac{\partial \mathbf{r}_k^+}{\partial \Delta t_k} \\ \frac{\partial \mathbf{v}_k^+}{\partial \Delta t_k} \\ 0 \\ 1 \end{bmatrix} \tag{67}$$

IV. Distance Constraint Derivative Computation

This section will discuss an operational constraint that imposes a minimum and/or maximum value on the distance between the spacecraft and bodies in the solar system, *i.e.*,

$$d_{\text{LB}} \leq r_{\text{s/c-body}} \leq d_{\text{UB}} \tag{68}$$

where d_{LB} and d_{UB} are defined by the analyst for each problem and for each body. For example, the spacecraft may be constrained to never get too close to the sun for thermal reasons, or may not be allowed to fly farther away from the earth than some maximum distance for communications reasons. These constraints occur often in real-world mission design, especially in low-thrust design where the desire to prevent the spacecraft from growing too hot is in conflict with the availability of more power and therefore more efficient propulsion closer to the sun.

The distance constraint is straightforward to pose in the optimization problem because it requires only looking up the position of the relevant solar system bodies at each time-step in the trajectory. However it is quite computationally expensive for two reasons. The first reason is that each ephemeris lookup requires an ephemeris database call, which is quite slow. The second reason is that computing analytical derivatives of the distance constraint requires recursive multiplications of the STMs and MTMs along the trajectory, similar to the STM-MTM chains used to compute the match point derivatives (e.g. Eq. (34)). In a large problem with many segments, over 50% of the execution time for the trajectory optimization is consumed by the derivative calculation code for the distance constraint.

In particular, the distance constraint and its derivatives must be computed at each maneuver point in the phase. The STM-MTM chains that must be computed in order to facilitate this are as follows:

$$\frac{\partial \mathbf{X}_j^-}{\partial \mathbf{X}_k^+} = \begin{cases} \mathbf{\Phi}_{k+1} & j = k+1 \\ \prod_{i=1}^{j-k} \left(\mathbf{\Phi}_{j+2-i} \mathbf{M}_{j+1-i} \right) \mathbf{\Phi}_{k+1} & 2 \leq k < j \leq N/2 \end{cases} \tag{69}$$

for forward half-phases, and

$$\frac{\partial \mathbf{X}_j^+}{\partial \mathbf{X}_k^-} = \begin{cases} \mathbf{\Phi}_{k+1} & j = k-1 \\ \prod_{i=j+2}^{k} \left(\mathbf{\Phi}_i \mathbf{M}_{i-1} \right) \mathbf{\Phi}_{k+1} & N/2+2 \leq j < k \leq N \end{cases} \tag{70}$$

for backward half-phases.

It should be noted that since the distance constraint is only enforced at maneuver locations along the phase, it is really only useful in conjunction with the MGALT model, which features a large number of impulses. In order to impose a similar constraint for the MGAnDSMs transcription, additional "non-maneuver" nodes can be introduced at regular intervals along the phase that are used exclusively to enforce the distance constraint.

V. Accuracy

The analytic techniques described in this work are not the only methods available for computing the constraint partials of a nonlinear program. Finite differencing methods are probably the simplest to implement, although this is typically accompanied by a decrease in program execution speed and worse a decrease in the accuracy of the partials. Other techniques such as complex step differentiation and algorithmic differentiation (AD) can produce near-machine precision derivative information, and are general techniques that do not require any *a priori* information about the

problem being solved. Both of these techniques typically require a longer runtime than specialized analytic formulae, if they are used to compute derivatives during a typical cost function evaluation. If they are used to compute only individual STMs and MTMs, which are then incorporated into derivative computations such as Eq. (29), then one would expect the runtimes to decrease. The interested reader is directed to a detailed study by Pellegrini and Russell, that examines finite differencing techniques of varying order, the complex step method as well as analytic methods as applied to STM computation for integrated trajectories [56]. This section will examine the accuracy of the partial derivative calculation methods described in this paper compared with the same derivatives obtained using central differencing as well as ones calculated using the AD library developed by Ghosh [38]. This AD software package uses C++ operator overloading in a tapeless forward-mode configuration.

An example 20 segment Earth to Mars MGALT trajectory phase was evaluated, and the Jacobian was computed using central differencing, the formulae presented in this paper as well as AD. The partial derivatives calculated with AD are assumed to be truth. An example column from the Jacobian is shown in Tables 4 and 5 for the finite differencing and analytic cases respectively. This column contains the partial derivatives the match-point continuity constraint $\mathbf{c}^\dagger$ with respect to the x component of the control vector from the first segment in the phase u_{x_1} (i.e. the segment furthest from the match-point in the forward half-phase). The error of the partial derivative values relative to the values computed using AD is shown for both methods.

Table 4. Example match-point derivatives computed using central differencing compared with algorithmic differentiation. A central differencing step size of 1.0e-7 was used.

Derivative	Value computed with FD	Error Relative to AD value
$\frac{\partial x^\dagger}{\partial u_{x_1}}$	-2020560.390983700	0.002538938555645e-5
$\frac{\partial y^\dagger}{\partial u_{x_1}}$	-166778.4769555778	0.136749253496038e-5
$\frac{\partial z^\dagger}{\partial u_{x_1}}$	-81633.22222039030	0.084320438683687e-5
$\frac{\partial \dot{x}^\dagger}{\partial u_{x_1}}$	-0.1006144971940313	0.016670205267864e-5
$\frac{\partial \dot{y}^\dagger}{\partial u_{x_1}}$	0.07758976470341465	0.029646679151652e-5
$\frac{\partial \dot{z}^\dagger}{\partial u_{x_1}}$	0.03673856948309212	0.050559366283141e-5
$\frac{\partial m^\dagger}{\partial u_{x_1}}$	2.346964682917494	0.027179713285220e-5

Table 5. Example match-point derivatives computed analytically compared with algorithmic differentiation.

Derivative	Value computed analytically	Error relative to AD value
$\frac{\partial x^\dagger}{\partial u_{x_1}}$	-2020560.339682915	0.0e-14
$\frac{\partial y^\dagger}{\partial u_{x_1}}$	-166778.7050242114	0.296659647101017e-14
$\frac{\partial z^\dagger}{\partial u_{x_1}}$	-81633.29105393921	0.267389352563618e-14
$\frac{\partial \dot{x}^\dagger}{\partial u_{x_1}}$	-0.1006144804213910	0.082758193947980e-14
$\frac{\partial \dot{y}^\dagger}{\partial u_{x_1}}$	0.07758974170063281	0.035772223244046e-14
$\frac{\partial \dot{z}^\dagger}{\partial u_{x_1}}$	0.03673855090831362	0.037774456163087e-14
$\frac{\partial m^\dagger}{\partial u_{x_1}}$	2.346964045019395	0.018921858253112e-14

As one would expect, the Jacobian values obtained with central differencing differ from the AD values with a relative error on the order of 1.0e-6 to 1.0e-8. With the analytic methods presented in this paper precision at, or very close to,

machine precision is achieved.

VI. Conclusion

This paper describes analytic methods for computing Jacobian matrix entries for two bounded-impulse trajectory models. The derivative calculations for the continuous-thrust model make accommodations for the accurate modeling of a propulsion unit powered using a solar electric system. These results are are general in the sense that they allow for the use of any solar electric power model constructed using smooth functions, and are also applicable for power sources such as radioisotope thermal electric generators. In addition, these techniques can also handle realistic operational constraints such as maintaining proximity bounds with respect to other bodies. The computations discussed in this work were verified for correctness using an algorithmic differentiation library.

Appendix

A. Two-Body Propagation

The transcriptions described in this paper utilize Keplerian two-body propagation using the Lagrange coefficients [48],

$$\begin{bmatrix} \mathbf{r}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = \begin{bmatrix} F & G \\ \dot{F} & \dot{G} \end{bmatrix} \begin{bmatrix} \mathbf{r}_k \\ \mathbf{v}_k \end{bmatrix}. \tag{71}$$

It should be noted that this could be replaced with numerical integration in the case of MGAnDSMs for the purpose of including additional dynamics thereby increasing the fidelity of the solution. For the case of MGALT, if the two-body propagation is replaced with numerical integration, the impulsive approximation should be discarded in favor of including the thrust term directly into the differential equations of motion [47].

It is generally beneficial to use a two-body propagation method capable of robustly propagating any orbit initial condition (i.e. elliptical, parabolic or hyperbolic) initialized by a search method. For this reason, a propagation method based on universal orbit variables is employed. Here we extend the propagator described by Der [57] to any conic orbit. The universal variables are defined according to the energy regime of the orbit. For elliptic trajectories:

$$\alpha = \frac{1}{a} = \frac{2}{r_k} - \frac{v_k^2}{\mu} > 0 \tag{72}$$

then defining

$$y = \alpha \chi^2 \tag{73}$$

$$C = \frac{1}{y}\left(1 - \cos(\sqrt{y})\right) \tag{74}$$

$$S = \frac{1}{y^3}\left(\sqrt{y} - \sin(\sqrt{y})\right) \tag{75}$$

the universal variables become:

$$\begin{cases} U_1 = \chi(1 - yS) \\ U_2 = \chi^2 C \\ U_3 = \chi^3 S \\ U_0 = 1 - \alpha U_2 \end{cases} \tag{76}$$

For a hyperbolic trajectory, $\alpha < 0$ and then

$$\begin{cases} U_0 = \cosh(\sqrt{-\alpha}\chi) \\ U_1 = \frac{1}{\sqrt{-\alpha}}\sinh(\sqrt{-\alpha}\chi) \\ U_2 = \frac{1}{\alpha}(1 - U_0) \\ U_3 = \frac{1}{\alpha}(\chi - U_1) \end{cases} \tag{77}$$

Parabolic trajectories are unlikely to be initialized by a search method. However, they can occur when an optimizer transitions a hyperbolic trajectory to an elliptic one, or vice versa. In this case, $\alpha = 0$ and

$$\begin{cases} U_0 = 1 \\ U_1 = \chi \\ U_2 = \frac{1}{2}U_1\chi \\ U_3 = \frac{1}{3}U_2\chi \end{cases} \tag{78}$$

Since it will never be the case that α will be exactly equal to zero (to machine precision), we find that a tolerance of $|\alpha| < 1 \times 10^{-12}$ works well.

The Lagrange coefficients and their time derivatives are given by:

$$F = 1 - \frac{U_2}{r_k} \tag{79}$$

$$\dot{F} = -\frac{\sqrt{\mu}}{r_{k+1} r_k} U_1 \tag{80}$$

$$\ddot{F} = -\frac{\sqrt{\mu}}{r_k} \left(\frac{\dot{U}_1}{r_{k+1}} - U_1 \frac{\dot{r}_{k+1}}{r_{k+1}^2} \right) \tag{81}$$

$$G = \frac{1}{\sqrt{\mu}} \left(r_k U_1 + \sigma_k U_2 \right) \tag{82}$$

$$\dot{G} = 1 - \frac{U_2}{r_{k+1}} \tag{83}$$

$$\ddot{G} = -\left(\frac{\dot{U}_2}{r_{k+1}} - U_2 \frac{\dot{r}_{k+1}}{r_{k+1}^2} \right) \tag{84}$$

where,

$$r_{k+1} = r_k U_0 + \sigma_k U_1 + U_2 \tag{85}$$

$$\dot{r}_{k+1} = r_k \dot{U}_0 + \sigma_k \dot{U}_1 + \dot{U}_2 \tag{86}$$

$$\sigma_{k+1} = \sigma_k U_0 + \left(1 - \alpha r_k \right) U_1 \tag{87}$$

$$\sigma_k = \frac{\mathbf{r}_k \cdot \mathbf{v}_k}{\sqrt{\mu}} \tag{88}$$

$$\frac{\partial U_0}{\partial \chi} = -\alpha U_1; \quad \frac{\partial U_n}{\partial \chi} = U_{n-1} \quad n = 1, 2, ... \tag{89}$$

$$\frac{\partial \chi}{\partial t} = \frac{\sqrt{\mu}}{r} \tag{90}$$

Kepler's equation in terms of the universal variable may be written as follows:

$$f = r_k U_1 + \sigma_k U_2 + U_3 - \sqrt{\mu} \Delta t \tag{91}$$

where,

$$\frac{\partial f}{\partial \chi} = r_k \tag{92}$$

$$\frac{\partial^2 f}{\partial \chi^2} = \sigma_k \tag{93}$$

Solution of Eq. (91) for the root χ is achieved by iteration using the Laguerre-Conway method [58], using the following update scheme:

$$\chi_{i+1} = \chi_i - \delta\chi; \quad \chi_0 = \begin{cases} \alpha\sqrt{\mu}\Delta t, & \text{if } \alpha > 0 \\ \frac{\sqrt{\mu}}{10 r_k}\Delta t, & \text{otherwise} \end{cases} \tag{94}$$

where,

$$\delta\chi \begin{cases} \frac{Nf}{\frac{\partial f}{\partial \chi} \pm \sqrt{\eta}}, & \text{if } \eta > 0 \\ \frac{f}{\frac{\partial f}{\partial \chi}}, & \text{otherwise} \end{cases} \tag{95}$$

and,

$$\eta = \left| (N-1)^2 \left(\frac{\partial f}{\partial \chi} \right)^2 - N(N-1) f \frac{\partial^2 f}{\partial \chi^2} \right| \tag{96}$$

Laguerre's root finding method, when applied to a general smooth function f, indicates that the sign in the denominator of Eq. (95) is positive if $f^{'}$ (Eq. (92) in this case) is non-negative and negative otherwise. Since Eq. (92) can never be negative for the particular case of Eq. (91), the sign will always be positive. The order N is typically set to 5, but can be increased if numerical instabilities are encountered.

B. Fundamental Perturbation STM

Vector expressions for the 3x3 perturbation STM quadrants:

$$\tilde{\mathbf{R}} = \frac{r_{k+1}}{\mu}(\mathbf{v}_{k+1}-\mathbf{v}_k)(\mathbf{v}_{k+1}-\mathbf{v}_k)^T + \frac{1}{r_k^3}\left[r_k(1-F)\mathbf{r}_{k+1}\mathbf{r}_k^T + C\mathbf{v}_{k+1}\mathbf{r}_k^T\right] + F\mathbb{I}_{3\times 3} \tag{97}$$

$$\mathbf{R} = \frac{r_k}{\mu}(1-F)\left[(\mathbf{r}_{k+1}-\mathbf{r}_k)\mathbf{v}_k^T - (\mathbf{v}_{k+1}-\mathbf{v}_k)\mathbf{r}_k^T\right] + \frac{C}{\mu}\mathbf{v}_{k+1}\mathbf{v}_k^T + G\mathbb{I}_{3\times 3} \tag{98}$$

$$\begin{aligned}\tilde{\mathbf{V}} = &-\frac{1}{r_k^2}(\mathbf{v}_{k+1}-\mathbf{v}_k)\mathbf{r}_k^T - \frac{1}{r_{k+1}^2}\mathbf{r}_{k+1}(\mathbf{v}_{k+1}-\mathbf{v}_k)^T \\ &+ F_t\left[\mathbb{I}_{3\times 3} - \frac{1}{r_{k+1}^2}\mathbf{r}_{k+1}\mathbf{r}_{k+1}^T \right. \\ &\left. + \frac{1}{\mu r_{k+1}}(\mathbf{r}_{k+1}\mathbf{v}_{k+1}^T - \mathbf{v}_{k+1}\mathbf{r}_{k+1}^T)\mathbf{r}_{k+1}(\mathbf{v}_{k+1}-\mathbf{v}_k)^T\right] \\ &- \frac{\mu C}{r_{k+1}^3 r_k^3}\mathbf{r}_{k+1}\mathbf{r}_k^T\end{aligned} \tag{99}$$

$$\mathbf{V} = \frac{r_k}{\mu}(\mathbf{v}_{k+1}-\mathbf{v}_k)(\mathbf{v}_{k+1}-\mathbf{v}_k)^T + \frac{1}{r_{k+1}^3}\left[r_k(1-F)\mathbf{r}_{k+1}\mathbf{r}_k^T - C\mathbf{r}_{k+1}\mathbf{v}_k^T\right] + G_t\mathbb{I}_{3\times 3} \tag{100}$$

In Equations (97) - (100), $\mathbb{I}$ is the 3x3 identity matrix. The quantity C is calculated using the universal functions:

$$\sqrt{\mu}C = 3U_5 - \chi U_4 - \sqrt{\mu}(t-t_0)U_2 \tag{101}$$

The functions U_n have traditionally been calculated with continued fraction expansions [48, 49], however, the following recursion relation can be used to compute higher-order universal functions:

$$U_n(\chi,\alpha) + \alpha U_{n+2}(\chi,\alpha) = \frac{\chi^n}{n!} \qquad n = 0,1,2,3,... \tag{102}$$

C. Patched-Conics Flyby Model Derivatives

Derivatives of the patched-conics flyby model constraints used by the MGALT and MGAnDSMs transcriptions are provided in Eq. (103)-(108)

$$\frac{\partial c_{\text{flyby-altitude}}}{\partial v^-_{\infty_x}} = -\frac{\mu \cos\left(\frac{\text{acos}(\alpha)}{2}\right)\left(v^+_{\infty_x} {v^-_{\infty_y}}^2 - v^-_{\infty_x} v^+_{\infty_y} v^-_{\infty_y} + v^+_{\infty_x} {v^-_{\infty_z}}^2 - v^-_{\infty_x} v^+_{\infty_z} v^-_{\infty_z}\right)}{r_{\text{flyby}}(\alpha-1)\left[1 - \frac{(v^-_{\infty_x} v^+_{\infty_x} + v^-_{\infty_y} v^+_{\infty_y} + v^-_{\infty_z} v^+_{\infty_z})^2}{\gamma\beta}\right]^{1/2} \gamma^{3/2}\beta^{3/2}} \tag{103}$$

$$\frac{\partial c_{\text{flyby-altitude}}}{\partial v^-_{\infty_y}} = -\frac{\mu \cos\left(\frac{\text{acos}(\alpha)}{2}\right)\left(v^+_{\infty_y} {v^-_{\infty_x}}^2 - v^-_{\infty_y} v^+_{\infty_x} v^-_{\infty_x} + v^+_{\infty_y} {v^-_{\infty_z}}^2 - v^-_{\infty_y} v^+_{\infty_z} v^-_{\infty_z}\right)}{r_{\text{flyby}}(\alpha-1)\left[1 - \frac{(v^-_{\infty_x} v^+_{\infty_x} + v^-_{\infty_y} v^+_{\infty_y} + v^-_{\infty_z} v^+_{\infty_z})^2}{\gamma\beta}\right]^{1/2} \gamma^{3/2}\beta^{3/2}} \tag{104}$$

$$\frac{\partial c_{\text{flyby-altitude}}}{\partial v^-_{\infty_z}} = -\frac{\mu \cos\left(\frac{\text{acos}(\alpha)}{2}\right)\left(v^+_{\infty_z} {v^-_{\infty_x}}^2 - v^-_{\infty_z} v^+_{\infty_x} v^-_{\infty_x} + v^+_{\infty_z} {v^-_{\infty_y}}^2 - v^-_{\infty_z} v^+_{\infty_y} v^-_{\infty_y}\right)}{r_{\text{flyby}}(\alpha-1)\left[1 - \frac{(v^-_{\infty_x} v^+_{\infty_x} + v^-_{\infty_y} v^+_{\infty_y} + v^-_{\infty_z} v^+_{\infty_z})^2}{\gamma\beta}\right]^{1/2} \gamma^{3/2}\beta^{3/2}} \tag{105}$$

where

$$\alpha = \frac{v^-_{\infty_x} v^+_{\infty_x} + v^-_{\infty_y} v^+_{\infty_y} + v^-_{\infty_z} v^+_{\infty_z}}{\beta^{1/2}\gamma^{1/2}}$$

$$\beta = {v^+_{\infty_x}}^2 + {v^+_{\infty_y}}^2 + {v^+_{\infty_z}}^2$$

$$\gamma = {v^-_{\infty_x}}^2 + {v^-_{\infty_y}}^2 + {v^-_{\infty_z}}^2$$

$$\frac{\partial c_{\text{flyby-altitude}}}{\partial v^+_{\infty_x}} = \frac{2v^+_{\infty_x}\,\mu}{r_{\text{flyby}}\xi^2} - \frac{2v^+_{\infty_x}\,\mu}{r_{\text{flyby}}\,\sin\left(\frac{\text{acos}(\epsilon)}{2}\right)\xi^2} - \frac{\mu \cos\left(\frac{\text{acos}(\epsilon)}{2}\right)\left(v^-_{\infty_x} {v^+_{\infty_y}}^2 - v^-_{\infty_y} v^+_{\infty_x} v^+_{\infty_y} + v^-_{\infty_x} {v^+_{\infty_z}}^2 - v^-_{\infty_z} v^+_{\infty_x} v^+_{\infty_z}\right)}{r_{\text{flyby}}(\epsilon-1)\left[1 - \frac{\phi^2}{\psi\xi}\right]^{1/2} \psi^{1/2}\xi^{5/2}} \tag{106}$$

$$\frac{\partial c_{\text{flyby-altitude}}}{\partial v^+_{\infty_y}} = \frac{2v^+_{\infty_y}\,\mu}{r_{\text{flyby}}\xi^2} - \frac{2v^+_{\infty_y}\,\mu}{r_{\text{flyby}}\,\sin\left(\frac{\text{acos}(\epsilon)}{2}\right)\xi^2} - \frac{\mu \cos\left(\frac{\text{acos}(\epsilon)}{2}\right)\left(v^-_{\infty_y} {v^+_{\infty_x}}^2 - v^-_{\infty_x} v^+_{\infty_y} v^+_{\infty_x} + v^-_{\infty_y} {v^+_{\infty_z}}^2 - v^-_{\infty_z} v^+_{\infty_y} v^+_{\infty_z}\right)}{r_{\text{flyby}}(\epsilon-1)\left[1 - \frac{\phi^2}{\psi\xi}\right]^{1/2} \psi^{1/2}\xi^{5/2}} \tag{107}$$

$$\frac{\partial c_{\text{flyby-altitude}}}{\partial v^+_{\infty_z}} = \frac{2v^+_{\infty_z}\,\mu}{r_{\text{flyby}}\xi^2} - \frac{2v^+_{\infty_z}\,\mu}{r_{\text{flyby}}\,\sin\left(\frac{\text{acos}(\epsilon)}{2}\right)\xi^2} - \frac{\mu \cos\left(\frac{\text{acos}(\epsilon)}{2}\right)\left(v^-_{\infty_z} {v^+_{\infty_x}}^2 - v^-_{\infty_x} v^+_{\infty_z} v^+_{\infty_x} + v^-_{\infty_z} {v^+_{\infty_y}}^2 - v^-_{\infty_y} v^+_{\infty_z} v^+_{\infty_y}\right)}{r_{\text{flyby}}(\epsilon-1)\left[1 - \frac{\phi^2}{\psi\xi}\right]^{1/2} \psi^{1/2}\xi^{5/2}} \tag{108}$$

where

$$\epsilon = \frac{\phi}{\psi^{1/2}\xi^{1/2}}$$
$$\xi = {v^+_{\infty_x}}^2 + {v^+_{\infty_y}}^2 + {v^+_{\infty_z}}^2$$
$$\phi = v^-_{\infty_x} v^+_{\infty_x} + v^-_{\infty_y} v^+_{\infty_y} + v^-_{\infty_z} v^+_{\infty_z}$$
$$\psi = {v^-_{\infty_x}}^2 + {v^-_{\infty_y}}^2 + {v^-_{\infty_z}}^2$$

Acknowledgments

The authors would like to thank Matthew Vavrina for his work developing the MGAnDSMs transcription and Dr. Alexander Ghosh at the University of Illinois for the use of his automatic differentiation library. In addition Dr. Englander would like to thank the internal research and development (IRAD) program and the planetary science new business office at NASA Goddard Space Flight Center and Dr. Ozimek would like to thank the Applied Physics Laboratory Civilian Space IR&D program.

References

[1] Prussing, J. E. and Chiu, J. H., "Optimal Multiple-Impulse Time-Fixed Rendezvous Between Circular Orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 1, 1986, pp. 17–22, doi:10.2514/3.20060.

[2] Vasile, M. and De Pascale, P., "Preliminary Design of Multiple Gravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 43, No. 4, 2006, pp. 794–805, doi:10.2514/1.17413.

[3] Conway, B., Chilan, C., and Wall, B., "Evolutionary principles applied to mission planning problems," *Celestial Mechanics and Dynamical Astronomy*, Vol. 97, No. 2, 2007, pp. 73 – 86, doi:10.1007/s10569-006-9052-7.

[4] Addis, B., Cassioli, A., Locatelli, M., and Schoen, F., "A global optimization method for the design of space trajectories," *Computational Optimization and Applications*, Vol. 48, No. 3, 2011, pp. 635–652, doi:10.1007/s10589-009-9261-6.

[5] Abdelkhalik, O. and Mortari, D., "N-Impulse Orbit Transfer Using Genetic Algorithms," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, 2007, pp. 456–459, doi:10.2514/1.24701.

[6] Vinkó, T. and Izzo, D., "Global Optimisation Heuristics and Test Problems for Preliminary Spacecraft Trajectory Design," Tech. Rep. GOHTPPSTD, European Space Agency, the Advanced Concepts Team, 2008. Available on line at www.esa.int/act.

[7] Englander, J., Conway, B., and Williams, T., "Automated Mission Planning via Evolutionary Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, 2012, pp. 1878–1887, doi:10.2514/1.54101.

[8] Sims, J. A. and Flanagan, S. N., "Preliminary Design of Low-Thrust Interplanetary Missions," in "AAS/AIAA Astrodynamics Specialist Conference, AAS Paper 99-338," Girdwood, Alaska, 1999.

[9] McConaghy, T. T., *GALLOP Version 4.5 User's Guide*, School of Aeronautics and Astronautics, Purdue University, 2005.

[10] Sims, J., Finlayson, P., Rinderle, E., Vavrina, M., and Kowalkowski, T., "Implementation of a Low-Thrust Trajectory Optimization Algorithm for Preliminary Design," in "AIAA/AAS Astrodynamics Specialist Conference, AIAA Paper 2006-6746," , 2006.

[11] Yam, C., di Lorenzo, D., and Izzo, D., "Constrained Global Optimization of Low-Thrust Interplanetary Trajectories," in "IEEE Congress on Evolutionary Computation (CEC)," Piscataway, NJ, 2010, pp. 1–7.

[12] Yam, C., di Lorenzo, D., and Izzo, D., "Low-Thrust Trajectory Design as a Constrained Global Optimization Problem," in "Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering," Vol. 225, 2011, pp. 1243–1251.

[13] Breakwell, J. V., "The Optimization of Trajectories," *Journal of the Society of Industrial and Applied Mathematics (SIAM)*, Vol. 7, No. 2, 1959, pp. 215–247, doi:10.1137/0107018.

[14] Lawden, D. F., *Optimal Trajectories for Space Navigation*, Butterworths Mathematical Texts, Butterworths, London, 1963.

[15] Lion, P. M. and Handelsman, M., "Primer Vector on Fixed-Time Impulsive Trajectories," *AIAA Journal*, Vol. 5, No. 1, 1968, pp. 127–132, doi:10.2514/3.4452.

[16] Merec, J.-P., *Optimal Space Trajectories*, Studies in Astronautics, Elsevier Scientific Publishing Company, 1979.

[17] Melbourne, W. G. and Sauer, J., C. G., "Optimum Interplanetary Rendezvous Trajectories with Power Limited Vehicles," *AIAA Journal*, Vol. 1, 1963, pp. 54–60, doi:10.2514/3.1468.

[18] Sauer, J., C. G., "Optimization of Multiple Target Electric Propulsion Trajectories," in "AIAA Paper 73-205," , 1975, doi:10.2514/6.1973-205.

[19] Jezewski, D. J., "Primer Vector Theory and Applications," Tech. Rep. TR-R454, NASA, 1975.

[20] Bryson, A. and Ho, Y., *Applied Optimal Control*, Taylor and Francis, 1975.

[21] Hull, D. G., *Optimal Control Theory for Applications*, Springer, New York, 2003.

[22] Enright, P. J. and Conway, B. A., "Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 981 – 985, doi:10.2514/3.20739.

[23] Enright, P. and Conway, B., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002, doi:10.2514/3.20934.

[24] Herman, A. and Conway, B., "Optimal Spacecraft Attitude Control Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 5, 1992, pp. 1287 – 1289, doi:10.2514/3.20983.

[25] Herman, A. L. and Conway, B. A., "Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592 – 599, doi:10.2514/3.21662.

[26] Herman, A. L. and Conway, B. A., "Optimal, Low-Thrust, Earth-Moon Orbit Transfer," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 141 – 147, doi:10.2514/2.4210.

[27] Tang, S. and Conway, B. A., "Optimization of Low-Thrust Interplanetary Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3, 1995, pp. 599 – 604, doi:10.2514/3.21429.

[28] Whiffen, G. J., "Static/Dynamic Control for Optimizing a Useful Objective," United States Patent No. 6 496 741, 2002. Filed March 25, 1999.

[29] Whiffen, G. J., "Mystic: Implementation of the Static Dynamic Optimal Control Algorithm for High-Fidelity, Low-Thrust Trajectory Design," in "AIAA/AAS Astrodynamics Specialist Conference and Exhibit, AIAA Paper 2006-6741," Keystone, Colorado, 2006, doi:10.2514/6.2006-6741.

[30] Lantoine, G. and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory," *Journal of Optimization Theory and Applications*, Vol. 154, No. 2, 2012, pp. 382–417, doi:10.1007/s10957-012-0039-0.

[31] Lantoine, G. and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 2: Application," *Journal of Optimization Theory and Applications*, Vol. 154, No. 2, 2012, pp. 418–442, doi:10.1007/s10957-012-0038-1.

[32] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Rev.*, Vol. 47, No. 1, 2005, pp. 99–131, doi:10.1137/S0036144504446096.

[33] Wächter, A. and Biegler, L. T., "On the Implementation of a Primal-Dual Interior Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57, doi:10.1007/s10107-004-0559-y.

[34] Büskens, C. and Wassel, D., *Modeling and Optimization in Space Engineering*, Springer New York, New York, NY, chap. The ESA NLP Solver WORHP, pp. 85–110, 2013, doi:10.1007/978-1-4614-4469-5_4.

[35] Karush, W., *Minima of Functions of Several Variables with Inequalities as Side Constraints*, Master's thesis, Chicago, 1939.

[36] Kuhn, H. W. and Tucker, A. W., "Nonlinear Programming," in "Proceedings of the 2^{nd} Berkeley Symposium," University of California Press, Berkeley, California, 1951, pp. 481–492. http://projecteuclid.org/euclid.bsmsp/1200500249.

[37] Lantoine, G., Russell, R. P., and Dargent, T., "Using Multicomplex Variables for Automatic Computation of High-Order Derivatives," *ACM Transactions on Mathematical Software*, Vol. 38, No. 3, 2012, p. 21 pages, doi:10.1145/2168773.2168774.

[38] Ghosh, A. R. M. and Coverstone, V., "Optimal cooperative CubeSat maneuvers obtained through parallel computing," *Acta Astronautica*, Vol. 107, 2015, pp. 130–149, doi:10.1016/j.actaastro.2014.10.042.

[39] Russell, R. P., "Primer Vector Theory Applied to Global Low-Thrust Trade Studies," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, 2007, pp. 460–472, doi:10.2514/1.22984.

[40] Englander, J. A., Vavrina, M., and Ghosh, A. R. M., "Multi-Objective Hybrid Optimal Control for Multiple-Flyby Low-Thrust Mission Design," in "AAS/AIAA Space Flight Mechanics Meeting, AAS Paper 15-227, Williamsburg, VA," , 2015.

[41] Vavrina, M., Englander, J. A., and Ghosh, A. R. M., "Coupled Low-Thrust Trajectory and Systems Optimization via Multi-Objective Hybrid Optimal Control," in "AAS/AIAA Space Flight Mechanics Meeting, AAS Paper 15-397, Williamsburg, VA," , 2015.

[42] Zimmer, S. and Ocampo, C., "Use of Analytical Gradients to Calculate Optimal Gravity-Assist Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 2, 2005, pp. 324–332, doi:10.2514/1.4825.

[43] Zimmer, S. and Ocampo, C., "Analytical Gradients for Gravity Assist Trajectories Using Constant Specific Impulse Engines," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 753–760, doi:10.2514/1.9917.

[44] Vavrina, M., Englander, J. A., and Ellison, D. H., "Global Optimization of N-Maneuver, High-Thrust Trajectories Using Direct Multiple Shooting," in "AIAA/AAS Space Flight Mechanics Meeting, AAS Paper 16-272, Napa Valley, CA," , 2016.

[45] Ellison, D. H., Conway, B. A., Englander, J. A., and Ozimek, M. T., "Partial Derivative Computation for Bounded-Impulse Trajectory Models Using Two-Sided Direct Shooting. Part 2: Application," *Journal of Guidance, Control, and Dynamics*, Vol. XX, No. XX, XXXX, pp. XXX–XXX.

[46] Prussing, J. and Conway, B., *Orbital Mechanics, Second Edition*, Oxford University Press, New York, 2012.

[47] Englander, J. A., Ellison, D. H., and Conway, B. A., "Global Optimization of Low-Thrust, Multiple-Flyby Trajectories at Medium and Medium-High Fidelity," in "AAS/AIAA Space Flight Mechanics Meeting, AAS Paper 14-309, Santa Fe, NM," , 2014.

[48] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, American Institute of Aeronautics and Astronautics Inc., Reston, Virginia, 1999.

[49] Shepperd, S. W., "Universal Keplerian State Transition Matrix," *Celestial Mechanics*, Vol. 35, No. 2, 1985, pp. 129–144, doi:10.1007/BF01227666.

[50] Goodyear, W. H., "A General Method for the Computation of Cartesian Coordinates and Partial Derivatives of the Two-Body Problem," Tech. Rep. cr-522, NASA, 1966.

[51] Sperling, H., "Computation of Keplerian Conic Sections," *American Rocketry Society*, Vol. 31, 1961, pp. 660–661.

[52] Herrick, S. H., "Universal Variables," *Astronomical Journal*, Vol. 70, 1965, pp. 309–315, doi:10.1086/109728.

[53] Lemmon, W. W. and Brooks, J. E., "A Universal Formulation for Conic Trajectories-Basic Variables and Relationships," Tech. Rep. 3400-601 9-tu000, TRW/Systems, Redondo Beach, CA, 1965.

[54] Pitkin, E. T., "Second transition partial derivatives via universal variables," *Journal of Astronautical Sciences*, Vol. 13, 1966, p. 204.

[55] Lantoine, G. and Russell, R. P., "A Fast Second-Order Algorithm for Preliminary Design of Low-Thrust Trajectories," in "59^{th} International Astronautical Congress, Paper IAC-08-C1.2.5, Glasgow, Scotland," , 2008.

[56] Pellegrini, E. and Russell, R. P., "On the Computation and Accuracy of Trajectory State Transition Matrices," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 11, 2016, pp. 2485–2499, doi:10.2514/1.G001920.

[57] Der, G. J., "An Elegant State Transition Matrix," in "AAS/AIAA Astrodynamics Conference, AIAA Paper 96-3660, San Diego, Ca," , 1996, doi:10.2514/6.1996-3660.

[58] Conway, B. A., "An Improved Method due to Laguerre for the Solution of Kepler's Equation," *Celestial Mechanics*, Vol. 39, No. 2, 1986, pp. 199–211, doi:10.1007/BF01230852.